ELMo 最好用的词向量《Deep Contextualized Word Representations》

mountain blue

做我爱的人的 baymax (●—●) forever

近年来，研究人员通过文本上下文信息分析获得更好的词向量。ELMo是其中的翘楚，在多个任务、多个数据集上都有显著的提升。所以，它是目前最好用的词向量，the-state-of-the-art的方法。这篇文章发表在**2018年的NAACL上，outstanding paper award。**下面就简单介绍一下这个“神秘”的词向量模型。

1. ELMo的优势

（1）ELMo能够学习到词汇用法的复杂性，比如语法、语义。

（2）ELMo能够学习不同上下文情况下的词汇多义性。

2. ELMo的模型简介

基于大量文本，ELMo模型是从深层的双向语言模型（deep bidirectional language model）中的内部状态(internal state)学习而来的，而这些词向量很容易加入到QA、文本对齐、文本分类等模型中，后面会展示一下ELMo词向量在各个任务上的表现。

3. 双向语言模型

语言模型就是生成文本的方式、方法，是多个 N 个词语的序列 $(t_1, t_2, \ldots, t_N)$ 的极大似然。前向语言模型就是，已知 $(t_1, t_2, \ldots, t_{k-1})$，预测下一个词语 t_k 的概率，写成公式就是

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \ldots, t_{k-1}).$$

最近，如《Exploring the limits of language modeling》、《On the state of the art of evaluation in neural language models》和《Regularizing and optimizing lstm language models》等论文中，首先使用character-level的RNN或CNN，计算得到“上下文无关”（context-independent）词向量表示 x_k^{LM}，然后将此向量feed进入L层的前向LSTM。在每一个位置 k，每个LSTM层会输出一个 $\vec{x}_k^{LM,j}$，其中j=1,....L. 最顶层的LSTM输出为 $\vec{x}_k^{LM,L}$，然后加上softmax来预测下一个词语 t_{k+1}。

既然是双向，后向的语言模型如下，即通过下文预测之前的 词语：

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, \ldots, t_N).$$

双向语言模型（biLM）将前后向语言模型结合起来，最大化前向、后向模型的联合似然函数即可，如下式所示：

$$\sum_{k=1}^{N} \left(log\, p(t_k | t_1, t_2, \ldots, t_{k-1}; \Theta, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + log\, p(t_k | t_{k+1}, t_{k+2}, \ldots, t_N; \Theta, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \right).$$

其中， Θ_x 和 Θ_s 分别是context-independent词向量训练时 和 soft max层的参数， $\overrightarrow{\Theta}_{LSTM}$ 和 $\overleftarrow{\Theta}_{LSTM}$ 则是双向语言模型的（前后向语言模型的）参数。

4. ELMo

ELMo是双向语言模型biLM的多层表示的组合，对于某一个词语 t_k ，一个L层的双向语言模型biLM能够由2L+1个向量表示：

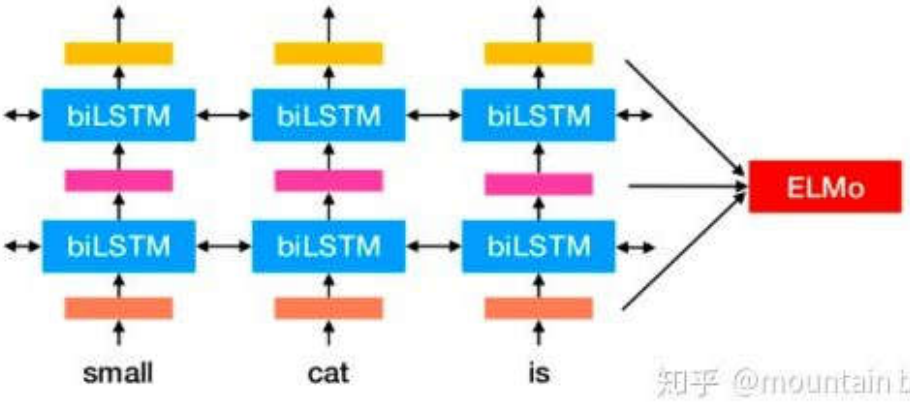
$$R_k = \{X^{LM}, \overrightarrow{h}_k^{LMj}, \overleftarrow{h}_k^{LMj} | j = 1, \ldots, L\} = \{h_k^{LMj}, | j = 1, \ldots, L\}$$

其中 $\{h_k^{LMj}\} = [\overrightarrow{h}_k^{LMj}; \overleftarrow{h}_k^{LMj}]$ 。

ELMo将多层的biLM的输出R整合成一个向量， $ELMo_k = E(R_k; \theta_e)$ 。最简单的情况是ELMo仅仅使用最顶层的输出，即 $E(R_k) = h_k^{LM,L}$ ，类似于TagLM和CoVe模型。但是，我们发现，最好的ELMo模型是将所有biLM层的输出加上normalized的softmax学到的权重 $s = Softmax(\mathbf{w})$ ：

$$E(R_k; \mathbf{w}, \gamma) = \gamma \sum_{j=0}^{L} s_j h_k^{LM,j}$$

其中 γ 是缩放因子。假如每一个biLM的输出具有不同的分布， γ 某种程度上来说相当于在weighting前对每一层biLM使用了layer nomalization。



上述ELMo词向量的运算过程基于RNN，但是ELMo词向量的运算不局限于RNN，CNN等也可以应用ELMo的训练。

5. 如何使用ELMo的词向量呢？
在supervised learning的情况下，可以各种自如的使用。。。

（1）直接将ELMo词向量 $ELMo_k$ 与普通的词向量 x_k 拼接（concat）[$x_k; ELMo_k$]。

（2) 直接将ELMo词向量 $ELMo_k$ 与隐层输出向量 h_k 拼接[$h_k; ELMo_k$]，在SNLI,SQuAD上都有提升。

6. ELMo模型的正则

ELMo模型中适当的dropout，或者将ELMo模型的weights加入 $\lambda||\mathbf{w} - \frac{1}{L+1}||_2^2$ 正则都会 imposes an inductive bias on the ELMo weights to stay close to an average of all biLM layers。

7. ELMo的效果

Textual entailment: stanford natural language inference (SNLI)数据集上提升了1.4%。

Question answering: 在stanford question answering dataset (SQuAD)数据集上提升了4.2%，将ELMo加入到之前的state-of-the-art的ensemble模型中，提升了10%。

Semantic role labeling: 比之前的state-of-the-art模型提高了3.2%，将ELMo加入到之前的state-of-the-art的单模型中，提升了1.2%。

Coreference resolution: 比之前的state-of-the-art模型提高了3.2%，将ELMo加入到之前的state-of-the-art的ensemble模型中，提升了1.6%。

Named entity extraction: 在CoNLL 2003 NER task数据机上提高了2.06%

Sentiment analysis: 比之前的state-of-the-art模型提高了3.3%，将ELMo加入到之前的state-of-the-art模型中，提升了1%。

8. 既然这么好用，哪里能“买“到呢？

这篇文章发表在**2018年的NAACL上，outstanding paper award**，本月初才发出，之前放在arxiv上，已有30+次引用。额，作者自己也觉得超好用，推出了工具包，可浏览项目主

页ELMo，github，paper。

pip install allennlp 即可享用。

The mind is not a vessel that needs filling, but wood that needs igniting. — Plutarch